Redis对事物的支持是部分支持。

## 1.1是什么

可以一次执行多个命令，本质是一组命令的集合。一个事务中的所有命令都会序列化，按顺序地串行化执行而不会被其它命令插入，不许加塞。

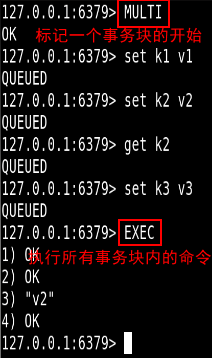
## 1.2能干嘛

一个队列中，一次性、顺序性、排他性的执行一系列命令。

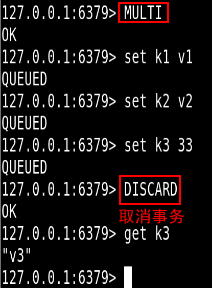
## 1.3怎么玩



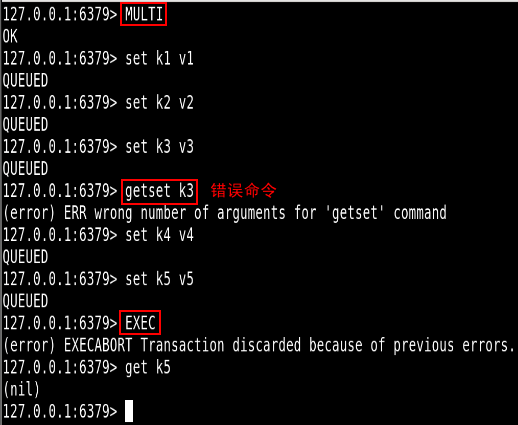
* case1：正常执行



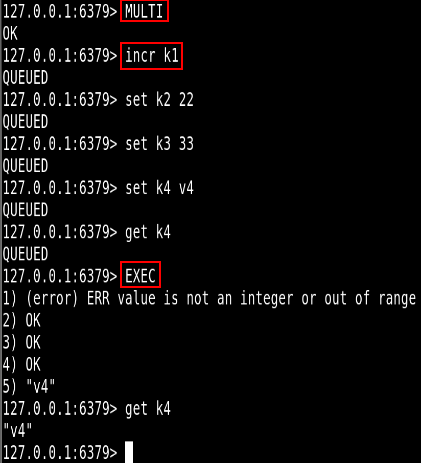
* case2：放弃事务



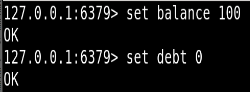
* case3：全体连坐



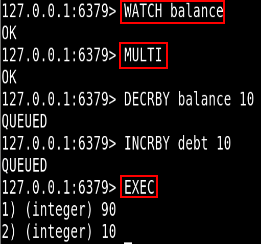
* case4：冤头债主



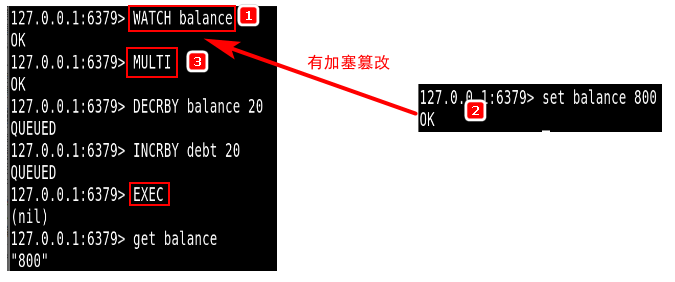
* case5：watch监控
  + 悲观锁/乐观锁/CAS（Check And Set）
    - 悲观锁：pessimistic lock，每次去拿数据的时候都认为别人会修改，所以每次在拿数据的时候都会上锁，这样别人拿数据就会block直到它拿到锁。传统的关系型数据库里边就用到了很多这种锁机制，比如行锁，表锁等，读锁，写锁等，都是在做操作之前先上锁。
    - 乐观锁（常用）：optimistic lock，每次去拿数据的时候都认为别人不会修改，所以不会上锁，但是在更新的时候会判断一下在此期间别人有没有去更新这个数据，可以使用版本号等机制。乐观锁适用于多读的应用类型，这样可以提高吞吐量。（乐观锁策略：提交版本必须大于记录当前版本才能执行更新）
  + 初始化信用卡可用余额和欠额



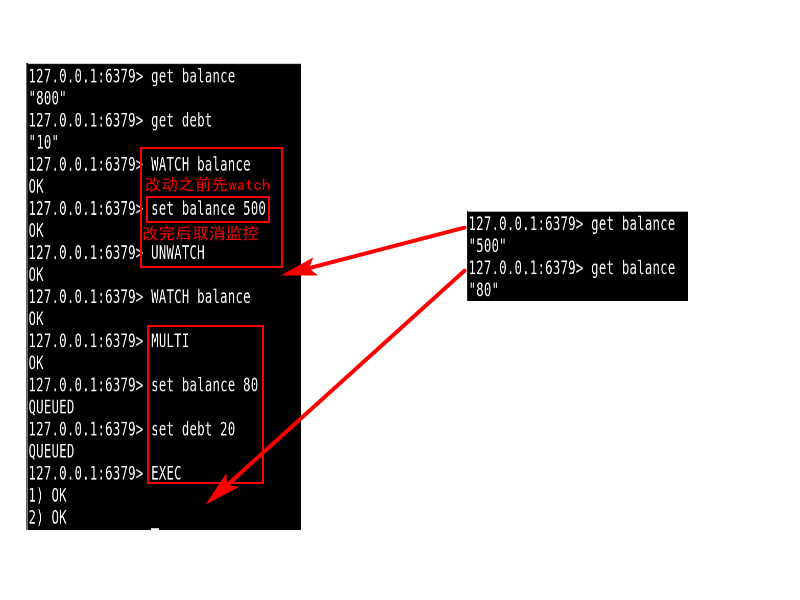
* + 无加塞篡改，先监控再开启multi，保证两笔金额变动在同一个事物内



* + 有加塞篡改



* + unwatch



* + 一旦执行了exec，之前加的监控锁都会被取消掉了
  + 小结：
    - watch指令，类似乐观锁，事务提交时，如果key的值已被别的客户端改变，比如某个list已被别的客户端push/pop过了，整个事务队列都不会被执行。
    - 通过watch命令在事务执行之前监控了多个keys，倘若在watch之后有任何key的值发生了变化，exec命令执行的事务都将被放弃，同时返回Nullmulti-bulk应答以通知调用者事务执行失败。

## 1.4三阶段

开启：以MULTI开始一个事务。

入队：将多个命令入队到事务中，接到这些命令不会立即执行，而是放到等待执行的事务队列里面。

执行：由EXEC命令触发事务。

## 1.5三特性

单独的隔离操作：事务中的所有命令都会序列化、按顺序地执行。事务在执行的过程中，不会被其他客户端发送来的命令请求所打断。

没有隔离级别的概念：队列中的命令没有提交之前都不会实际的被执行，因为事务提交前任何指令都不会被实际执行，也就不存在”事务内的查询要看到事务里的更新，在事务外查询不能看到”这个让人万分头痛的问题。

不保证原子性：redis同一个事务中如果有一条命令执行失败，其后的命令仍然会被执行，没有回滚。